Rod Constraints for Simplified Ragdolls

Chris Lewin[†‡*] Matt Thorman[‡] Tom Waterson[‡] Chris Williams[†] Phil Willis[†]

Centre for Digital Entertainment, University of Bath[†] Electronic Arts[‡]

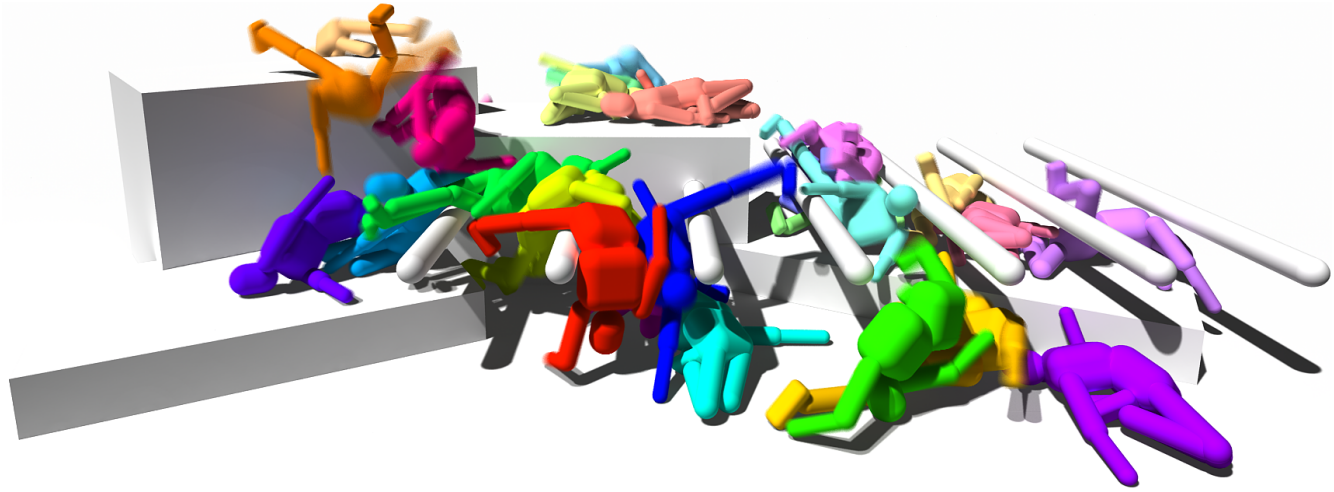

Figure 1: *30 simplified ragdoll fall together into a complex, compact physics world.*

Abstract

Physics-based animation has become a standard feature in modern games. Typically, the bones in a character's animation rig are each associated with a simulated rigid body, leading to a jointed assembly commonly called a *ragdoll*. The high density of animation bones in the spine area can cause instability and performance issues, so we are motivated to find a simplified physical representation for this region.

We approximate the spine region of a ragdoll as an inextensible elastic curve, building a circular arc constraint based on the Kirchhoff rod model. Our simplified spine shows improved performance and stability over the standard group of socket joints, and proves to be more controllable. To model general elastic rods we use soft position constraints in place of forces, leading to a stable maximal coordinate formulation of inextensible Kirchhoff rods.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation;

1 Introduction

While physics-based animation holds a place of increasing importance in games, several factors restrict its wider use. While any performance cost can be quantified and accounted for, simulation instability and a lack of controllability present unknown quantities to game designers, who as a result are often tempted to reduce the impact the simulation can have on gameplay. Improving stability is thus an important goal if we wish to see wider adoption of physical animation in the games industry.

Ragdoll simulation is one of the oldest uses of physics in games, pre-dating even the widespread use of rigid body solvers [Jakobsen 2001]. Although multi-body systems research in academia often assumes the use of a reduced coordinate *dynamics algorithm* such as Featherstone's algorithm [Featherstone 1987], it is more common in games to use a constrained rigid body simulator such as Bullet [Coumans 2005]. This *maximal coordinate* approach has the advantage of allowing characters to interact with other rigid bodies with no extra effort, unlike reduced coordinate methods which require additional constraints to link the two simulations together. However, the sequential impulse-type solvers used by many game physics engines converge slowly when faced with ragdoll assemblies, whose connectivity graphs have a deep tree topology. Because game engines typically use a fixed number of iterations, this slow convergence manifests as residual joint errors and jittering.

The simplest remedy for this situation is to decrease the number of bones in the animation rig. However, this is usually not viable because authored animation needs large numbers of bones in certain areas to be expressive. We could take smaller timesteps or use more solver iterations, but this has a severe performance impact. If we want to have both expressive animation and simple ragdolls, we need to break the isomorphism between the physics and animation skeletons.

To this end, we define a *simplified* ragdoll as one that does not have a 1:1 mapping between animation bones and physics bodies. The goal then becomes to find *simplified regions*: areas of the skeleton in which there is a simple approximation to the underlying kinematics. The results of the simple simulation are then interpolated back to the associated animation bones. The spine region is a good candidate for simplification because the underlying kinematics of the spine are similar to those of an inextensible rod. We can capture these dynamics by building an inextensible *rod constraint*.

2 Related Work

Simulation of elastic rods is a popular field, so we will only cover the most closely related work here. The reader is referred to the recent survey paper [Ward et al. 2007] for a more comprehensive

*clewin@ea.com

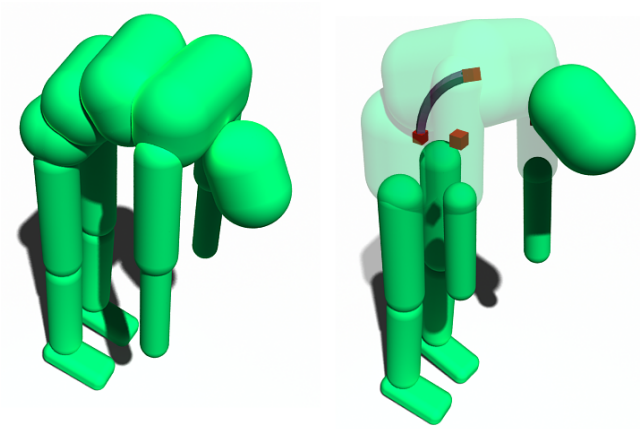

Figure 2: *Comparison between a standard ragdoll with three joints in the spine (left) and a simplified ragdoll with a single rod constraint (right), both slumping forward under gravity.*

view of the field as it pertains to computer graphics. Maximal coordinate methods based on mass-spring networks [Selle et al. 2008] and shape matching [Rungjiratananon et al. 2012] can be very fast and can capture bending and twisting dynamics effectively, but have difficulty preserving length.

Other researchers adopt reduced coordinate approaches. Bergou et al. [2008] perform a comprehensive discretisation of Langer and Singers work [1996] on Kirchhoff rods, allowing them to accurately simulate instability phenomena such as the buckling of elastic rings. Bertails et al. [2006] use helical elements of constant curvature and torsion to model curled hair wisps. Chains of rigid bodies [Hadap and Magnenat-Thalmann 2001] have also been used effectively. Methods that use the control points of spline curves as reduced coordinates [Remion et al. 1999; Nocent and Remion 2001] have been used to accurately model rod-like mechanical parts [Theetten et al. 2008] as well as knitted cloth [Kaldor et al. 2008]. Discounting collision detection, these methods are not overly expensive. However, for reasons discussed previously we prefer not to use reduced coordinate methods.

There is little academic work on ragdoll construction. Techniques which synthesize motion at runtime such as [Yin et al. 2007] have no need to interpolate from (and extrapolate back to) an authored animation skeleton, and thus can use ragdolls that are as simple as possible. However, much work has been done on alternative joint representations for inverse kinematics. Lee and Terzopolous [2008] developed a joint model based on spline surfaces, with the spline parameters as the reduced coordinates of the simulation. This method allowed them to model complex joints, although it would be prohibitively expensive to use in a maximal coordinate setting. Engell et al. [2012] proposed a data-driven approach based on distance fields in angle-space, allowing them to model the complex constraint manifold of the shoulder joint with constant-time constraint projections, at the cost of a large memory footprint.

3 Contributions

We show how to model inextensible elastic rods as constraints between rigid body pairs. We define the rods shape implicitly by the position and orientation of the two bodies, imposing an additional constraint to ensure the resulting shape is convenient - that of a circular arc. We emulate the bending and twisting forces using soft constraints, allowing stiff rods to be stable even when using long timesteps.

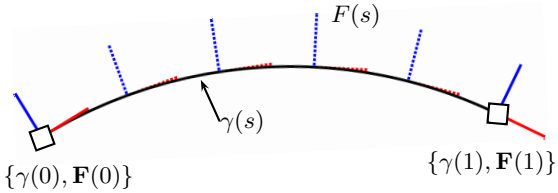

Figure 3: *Adapted framed curve between rigid bodies.*

We improve the convergence behaviour of character ragdolls by replacing the many joints in the spine region with our rod constraint. This leads to significantly reduced joint violation when using a fixed number of iterations, as well as improved controllability.

4 Background

We adopt the uniform variation of the Kirchhoff elastic rod model [Langer and Singer 1996], which describes a rod Γ as a centerline curve $\gamma(s)$ and an orthonormal material frame $F(s) = \{\mathbf{t}(s), \mathbf{m}_1(s), \mathbf{m}_2(s)\}$, representing the rotation of the rod's cross-section about the centerline. The material frame is called *adapted* because $\mathbf{t}$ is tangent to the centerline: $\mathbf{t}(s) = \gamma(s)' = \frac{d}{ds}\gamma(s)$. Because we assume inextensibility $s \in [0, 1]$ must be an arc length parameterisation, and $|\mathbf{t}(s)| \equiv 1$.

The elastic energy of the rod is related to the bending and twisting strains:

$$E(\Gamma) = \frac{1}{2}\int k_b \kappa^2 \, ds + \frac{1}{2}\int k_t \tau^2 \, ds, \tag{1}$$

where $\kappa = |\mathbf{t}'|$ is the bending strain or *curvature* of the centerline and τ is the twisting strain, which is the angle between the material frame and the *natural frame* of the centerline. The natural frame is one of many possible framings of a space curve, with the distinguishing factor that it has no inherent twist. Because we assume the rod is inextensible, there is no stretching component to the energy.

While a full discussion of rigid body physics is beyond the scope of this paper (see [Bender et al. 2012] for a recent survey), we should still make some notation clear. We consider a rigid body i to be an oriented particle with coordinates $\mathbf{x}_i = \{\mathbf{p}_i, \mathbf{q}_i\}$, where $\mathbf{p}_i$ is the position of its center of mass and $\mathbf{q}_i$ is a quaternion representing its orientation. Each body is equipped with a mass m_i and inertia $\mathbb{I}_i$. We use an impulse-based method [Bender and Schmitt 2006], solving the system from a set of preview coordinates $\mathbf{x}^*$. We deal with collision detection and contact resolution in standard ways, and as they are orthogonal to the purpose of this paper we will not discuss them here.

5 Our Method

To build a rod constraint between the rigid bodies $\mathbf{x}_a$ and $\mathbf{x}_b$, we need the ingredients of a rod in the Kirchhoff model: a vector-valued curve function $\gamma(s, \mathbf{x}_a, \mathbf{x}_b)$ and a quaternion-valued material frame function $\mathbf{F}(s, \mathbf{x}_a, \mathbf{x}_b)$. We specify that each end of the rod is attached to a rigid body at its center of mass (Figure 3):

$$\begin{aligned} \gamma(0) = \mathbf{p}_a, \gamma(1) = \mathbf{p}_b, \\ \mathbf{F}(0) = \mathbf{q}_a \mathbf{F}_a^{\text{bod}}, \mathbf{F}(1) = \mathbf{q}_b \mathbf{F}_b^{\text{bod}}, \end{aligned} \tag{2}$$

where $\mathbf{F}_i^{\text{bod}}$ is the orientation of the rod in body i's local space. The *body tangents* $\mathbf{t}_i$ and *body normals* $\mathbf{n}_i$ are then defined as directions in $\mathbf{F}_i^{\text{bod}}$.

Given a specific form for the curve $\gamma(s)$, we can constrain its length and find soft constraint equivalents of the bending and twisting

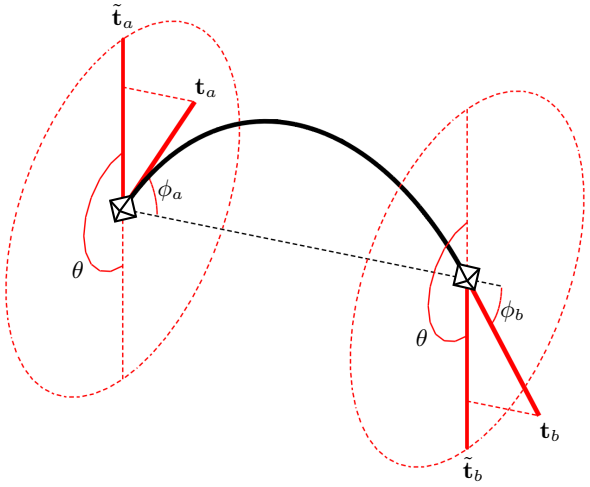

Figure 4: *Visualisation of the arc constraint. The angle θ between the tangents' rejections onto $\hat{\mathbf{o}}$ ($\tilde{\mathbf{t}}_a$ and $\tilde{\mathbf{t}}_b$) must be equal to π, and the angles ϕ_a and ϕ_b must be equal and opposite.*

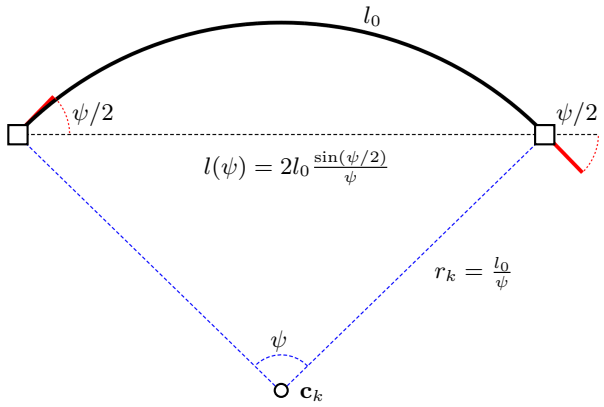

Figure 5: *Circular arc in the plane of curvature. Assuming the rod is the correct length l_0, the length of the offset l can be determined as a function of ψ.*

forces. Our constraint equations are scalar or vector functions of the coordinates of rigid body pairs $\mathbf{C}(\mathbf{x}_a, \mathbf{x}_b)$ that are equal to zero when the constraint is satisfied.

5.1 Length constraint

To constrain the length of the curve $\gamma(s)$, we need to compare its arc length to the length of the undeformed rod l_0:

$$C_{\text{len}}(\mathbf{x}_a, \mathbf{x}_b) = \int_0^1 |\gamma'(s, \mathbf{x}_a, \mathbf{x}_b)| \, ds - l_0. \tag{3}$$

To enforce this constraint we need to be able to calculate the arc length of γ, which in general is a hard problem. While we can numerically obtain an arc length parameterisation for any curve, the amount of computation required would render the constraint unsuitable for realtime use. This problem must inform our choice of γ.

5.2 Elastic constraints

Instead of using forces to simulate the bending and twisting behaviour of the rod, we will use soft constraints. This is a technique used to great effect in the field of *position-based dynamics* [Müller et al. 2007]. The major advantage over force-based methods is guaranteed stability: increasing the strength of the constraint will not create numerical stiffness. To use this technique we take the energy terms as constraint equations:

$$C_{\text{bend}}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{2}\alpha_b \int_0^1 \kappa^2 \, ds, \tag{4}$$

$$C_{\text{twist}}(\mathbf{x}_a, \mathbf{x}_b) = \frac{1}{2}\alpha_t \int_0^1 \tau^2 \, ds. \tag{5}$$

Note that instead of bending and twisting stiffnesses k_b and k_t we use equivalent relaxation factors α_b and α_t. In position-based dynamics we make this substitution because our constraints will be solved as velocity updates rather than forces. True stiffness constants have a range of 0 (completely soft) to infinity (completely rigid). Using specific relaxation factors rescales this range from 0 to 1.

The major disadvantage of this approach is a loss of physical realism. In particular, using soft constraints will cause damping of the resulting oscillations. For games this is not a big problem, as stability is valued over physical realism. Another issue encountered by Müller et al. [2007] is that it is difficult to estimate the effect of a given specific relaxation when compounded over many sequential iterations. In their case they were able to find a suitable rescaling, but for more complex constraints this may not be possible.

5.3 Choice of curve

Note that we have left the curve function $\gamma(s, \mathbf{x}_a, \mathbf{x}_b)$ undefined. The expressiveness and efficiency of this constraint hinge strongly on our choice of curve type. We require a curve that is fully determined by the body positions and frames, and which has an easily-calculated arc length. One might consider following Theetten et al. [2008] and using a cubic spline curve to form γ. This passes the first test but fails the second, as expressions for the arc length of cubic splines are complex.

We thus instead follow Bertails et al. [2006] and choose to restrict our curves to those with easily-obtainable expressions for arclength, curvature and torsion. While they chose helical segments, which have constant curvature and constant Frenet torsion, we choose circular arcs, which have constant curvature and zero torsion. This choice satisfies the second condition, but poses some problems for the first. While a unique cubic Hermite spline exists for any set of coordinates $\mathbf{x}_a$ and $\mathbf{x}_b$, this is not true for circular arcs. We need to include an additional constraint on the two bodies that will force them into an admissible configuration - one in which we can draw a valid circular arc between them.

5.4 Curve constraint

To constrain a rigid body pair such that they have an adjoining circular arc, we need to find sufficient conditions on the positions and orientations of the bodies at each end of the curve. We do this by inspecting a valid arc (Figure 4). We note that $\mathbf{t}_a, \mathbf{t}_b$ and $\hat{\mathbf{o}}$ must all lie in the same plane, which means that the angle θ must be equal to π. We also note that the angles ϕ_a and ϕ_b must be equal and opposite. Effectively, these conditions require that $\mathbf{t}_b$ be the *reflection* of $\mathbf{t}_a$ about $\hat{\mathbf{o}}$, and we can encode this using a similar equation to specular reflection about a surface normal:

$$\mathbf{C}_{\text{curve}}(\mathbf{x}_a, \mathbf{x}_b) = \mathbf{t}_a + \mathbf{t}_b - 2(\mathbf{t}_a \cdot \hat{\mathbf{o}})\hat{\mathbf{o}}. \tag{6}$$

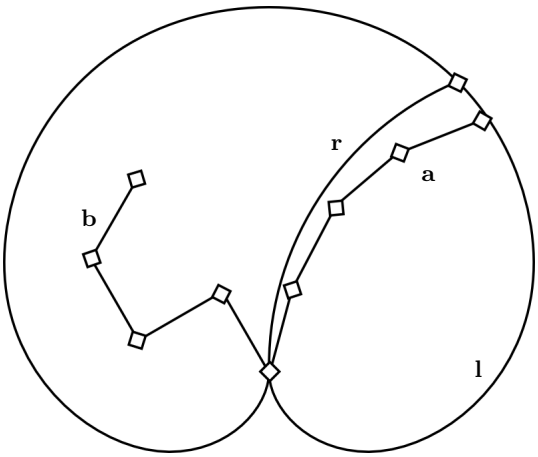

Figure 6: *Comparison of the kinematics of chains of ball and socket joints and a single rod constraint. If each joint in a chain makes the same angle (chain* **a***), then the locus of points the end of the chain can achieve (curve* **l***) is similar to that of the rod constraint (curve* **r***). If this is not true (chain* **b***) then the assumption breaks down.*

5.5 Specific form of length and elastic constraints

When this constraint is satisfied we can look at the bodies' configuration in the plane of curvature (Figure 5) to find simple expressions for the length preservation and elastic constraints. We can define $\gamma(s)$ as a sweep about the center of curvature $\mathbf{c}_k$ from $\mathbf{p}_a$ to $\mathbf{p}_b$:

$$\gamma(s, \mathbf{x}_a, \mathbf{x}_b) = \mathbf{c}_k + \tilde{r}(s)(\mathbf{p}_a - \mathbf{c}_k), \tag{7}$$

where $\tilde{r}(s)$ is the matrix form of an axis-angle rotation about the plane normal:

$$r(s, \mathbf{x}_a, \mathbf{x}_b) = \psi s \hat{\mathbf{n}}, \tag{8}$$

where $\psi(\mathbf{x}_a, \mathbf{x}_b) = \arccos(\mathbf{t}_a \cdot \mathbf{t}_b)$ is the curve bending angle. We take the approach of correcting the length purely through linear motion along the offset axis; thus assuming a fixed bend angle ψ we can derive the correct length of the offset for a given rest length l_0:

$$C_{\text{len}}(\mathbf{x}_a, \mathbf{x}_b) = |\mathbf{p}_b - \mathbf{p}_a| - 2l_0 \frac{\sin(\psi/2)}{\psi}. \tag{9}$$

With this constraint satisfied, we notice the radius of curvature $r_k = |\mathbf{c}_k - \mathbf{p}_a| = l_0/\psi$. Thus the curvature is constant over the rod:

$$\kappa = \frac{1}{r_k} = \frac{\psi}{l_0}, \tag{10}$$

and the bending constraint is

$$C_{\text{bend}}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\alpha_b \psi^2}{2l_0^2}. \tag{11}$$

We recall that the twist τ is the angle between the natural and material frames. We know the material frames at the beginning and end of the circular arc: $\mathbf{F}(0) = \mathbf{q}_a \mathbf{F}_a^{\text{bod}}$, $\mathbf{F}(1) = \mathbf{q}_b \mathbf{F}_b^{\text{bod}}$. Thus we can find the total angle between the two frames and subtract the bending angle, leaving us with the twisting angle:

$$\tau = \xi - \psi, \tag{12}$$

where ξ is the total angle between the quaternions $\mathbf{F}(0)$ and $\mathbf{F}(1)$. The twisting constraint is thus:

$$C_{\text{twist}}(\mathbf{x}_a, \mathbf{x}_b) = \frac{\alpha_t \tau^2}{2}. \tag{13}$$

6 Ragdoll Considerations

Although our rod constraint has a range of motion similar to a group of ball and socket joints, the two are not interchangeable. Specifically, if the internal angles of the ball and socket chain are very different to one another then the end of the chain can take very different positions to the end of the rod constraint (Figure 6). The rod approximation is thus closest to being valid when there are a large number of joints in the chain, and the internal angles are all the same. Fortunately, this is usually close to the truth in animations authored for games.

However, most animations do deviate by some amount from this ideal state. A very important case to consider for ragdolls is that a character is initially animated kinematically, and some action from the player causes it to transition to a ragdoll. This means we need to allow any configuration of the ragdoll given to us by a source animation. We can deal with this by instantiating the rod only when the transition occurs, setting its rest length and body frames such that the new constraint is valid.

Finally, we need to interpolate the results of the simplified simulation back to intermediate animation bones that do not have associated physics bodies. We can do this by linearly interpolating the position or orientation changes of the rigid body at the top of the spine, sharing them out between the animation bones. If we use the orientation changes only, the bone lengths will be conserved but the end of the chain of animation bones will end up in a different position to the rigid body (this effect gets worse if the spine is severely kinked). If we also interpolate the position changes, the bone lengths will not be conserved but the positions will match. We have found using orientations only tends to produce fewer artifacts in practice.

7 Implementation

Although our rod model consists of several constraint equations, none of these are particularly useful individually. By solving all the constraints together in one function, we can share some calculations common to each and end up doing less work overall. The code accompanying this paper includes a sample implementation of the rod constraint.

Our solver and constraint functions are implemented in our physics engine as branch-free SIMD code, solving four constraints at once. It takes on average 60 ns to solve one rod joint on a 2.4GHz Intel Xeon CPU, while the equivalent ball and socket joint takes 20ns on average. We thus break even on efficiency if we replace three ball and socket joints with a single rod. Because typical animation rigs for human characters use four joints between the hips and upper spine, this translates to a slight performance improvement in practice. This does not tell the whole story, however: if the number of iterations we use was kept artificially high by the complexity of our ragdolls, then simplifying them will remove a performance bottleneck by allowing us to use fewer iterations.

8 Results

Because game physics engines typically use a fixed number of iterations, slow convergence of the constraint solver manifests as residual error at the end of each frame. It thus makes sense to look at this metric when comparing methods. Figure 8 shows that there is a dramatic improvement in residual error between a standard and simplified ragdoll in a passive bending scene, and a smaller but still consistently positive difference in a more natural scene that includes collisions.

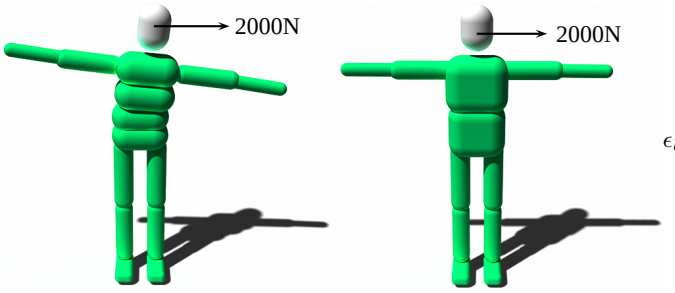

Figure 7: *Comparison of controllability between a ragdoll with three joints in the spine (left) and a single rod constraint (right). With the hips fixed, the characters try to maintain their initial pose in the face of an external force.*

This improvement in convergence speed is particularly important when we try to control the ragdoll. In methods such as PD control, the kinematic depth of the ragdoll has a major effect on the ability of a character to follow a target animation. As a crude metric for controllability, we can look at each ragdoll's ability to maintain a target pose using internal torque constraints in the face of external forces. Figure 7 shows each type of ragdoll's response to such a force. Even though the constraints driving the ragdoll to the target pose have infinite strength, the large number of joints in the spine causes the left ragdoll to converge to the target pose very slowly, resulting in a noticeable deviation after 20 solver iterations. Reducing the number of joints in the spine allows forces to propagate through the bottom ragdoll in fewer iterations, leading to a greater ability to maintain the target pose.

It should be noted that our ball and socket joints are solved using simplified methods that are cheap but converge slowly, and thus the error behaviour is not directly comparable to other published methods that are more accurate. However, our implementation of the rod constraint also uses cheap approximations. It should always be possible to get significant benefits from reducing the number of joints in the spine if one replaces a group of ball and socket constraints with a single similarly accurate rod constraint.

It should also be mentioned that one could achieve the same improvements in convergence behaviour by simply reducing the number of ball and socket joints in the spine to one. However, a single joint of this type has very different kinematics to a chain of joints, and the resulting motion looks unnatural. Using our rod constraint allows the ragdoll to both converge quickly and look plausible.

9 Limitations and Future Work

The restriction of the constraint curve to a circular arc is useful for keeping control of our ragdolls, but it does not lend itself to a truly expressive model of Kirchhoff elastic rods. An interesting extension would be to follow [Bertails et al. 2006] more closely and use helical constraint curves. This would mean finding a set of constraint equations that could project the configuration of two arbitrary rigid bodies to the nearest configuration with a valid adjoining helix.

The other major candidate for simplification in ragdolls is the shoulder region. Unlike the spine, the shoulder does not have a simple kinematic approximation. We believe the problem of finding an expressive shoulder constraint is best approached in a data-driven way, similar to that taken by Engell et al. [2012], and we plan to work on this soon.

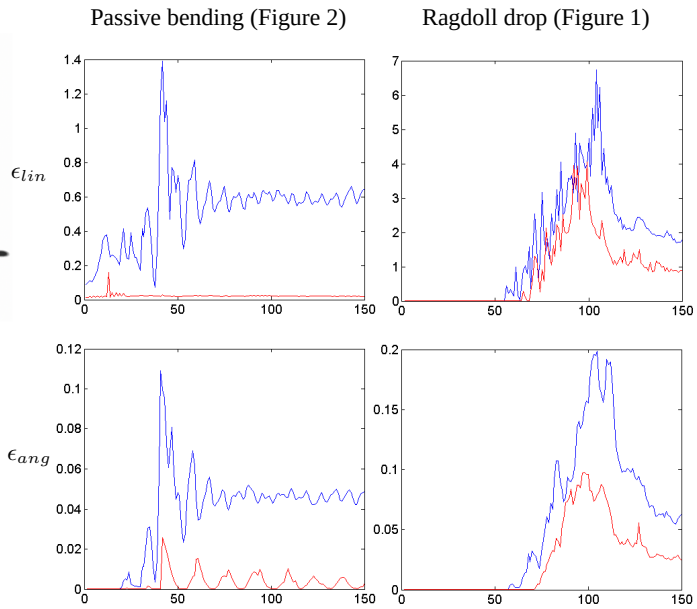

Figure 8: *Residual error per frame after 20 iterations in ragdolls using spines with three joints (blue) and our rod constraint (red). The top graphs show the total dislocation of the joints in each ragdoll in cm, and the bottom graphs show the total angular violation in radians.*

Acknowledgements

The authors would like to thank Darren Cosker, Tom Nugent and the whole EAPhysics team. This work was supported by the RCUK Digital Economy programme. The rendering for this paper was achieved with Pixie (www.renderpixie.com), which was written by Okan Arikan.

References

BENDER, J., AND SCHMITT, A. 2006. Fast dynamic simulation of multi-body systems using impulses. In *Virtual Reality Interactions and Physical Simulations (VRIPhys)*, 81–90.

BENDER, J., ERLEBEN, K., TRINKLE, J., AND COUMANS, E. 2012. Interactive simulation of rigid body dynamics in computer graphics. In *EUROGRAPHICS 2012 State of the Art Reports*, Eurographics Association.

BERGOU, M., WARDETZKY, M., ROBINSON, S., AUDOLY, B., AND GRINSPUN, E. 2008. Discrete Elastic Rods. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, vol. 27, 63:1–63:12.

BERTAILS, F., AUDOLY, B., CANI, M.-P., QUERLEUX, B., LEROY, F., AND LÉVÊQUE, J.-L. 2006. Super-helices for predicting the dynamics of natural hair. In *ACM SIGGRAPH 2006 papers*, ACM, New York, NY, USA.

COUMANS, E., 2005. Bullet physics. http://bulletphysics.org.

ENGELL-NRREGRD, M., NIEBE, S., AND ERLEBEN, K. 2012. A joint-constraint model for human joints using signed distance-fields. *Multibody System Dynamics 28*, 69–81.

FEATHERSTONE, R. 1987. *Robot Dynamics Algorithms*. Kluwer Academic Publishers, Norwell, MA, USA.

HADAP, S., AND MAGNENAT-THALMANN, N. 2001. Modeling dynamic hair as a continuum. *Computer Graphics Forum 20*, 3, 329–338.

JAKOBSEN, T. 2001. Advanced character physics. In *Game Developer's Conference 2001*.

KALDOR, J. M., JAMES, D. L., AND MARSCHNER, S. 2008. Simulating knitted cloth at the yarn level. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 65:1–65:9.

LANGER, J., AND SINGER, D. A. 1996. Lagrangian aspects of the kirchhoff elastic rod. *SIAM Rev. 38*, 4 (Dec.), 605–618.

LEE, S.-H., AND TERZOPOULOS, D. 2008. Spline joints for multibody dynamics. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 22:1–22:8.

MÜLLER, M., HEIDELBERGER, B., HENNIX, M., AND RATCLIFF, J. 2007. Position based dynamics. *J. Vis. Comun. Image Represent. 18*, 2 (Apr.), 109–118.

NOCENT, O., AND REMION, Y. 2001. Continuous deformation energy for dynamic material splines subject to finite displacements. In *Proceedings of the Eurographic workshop on Computer animation and simulation*, Springer-Verlag New York, Inc., New York, NY, USA, 88–97.

REMION, Y., NOURRIT, J.-M., AND GILLARD, D. 1999. Dynamic animation of spline like objects. In *Proc. WSCG'99*, V. Skala, Ed.

RUNGJIRATANANON, W., KANAMORI, Y., METAAPHANON, N., BANDO, Y., CHEN, B.-Y., AND NISHITA, T. 2012. Animating strings with twisting, tearing and flicking effects. *Comput. Animat. Virtual Worlds 23*, 2 (Mar.), 113–124.

SELLE, A., LENTINE, M., AND FEDKIW, R. 2008. A mass spring model for hair simulation. In *ACM SIGGRAPH 2008 papers*, ACM, New York, NY, USA, SIGGRAPH '08, 64:1–64:11.

THEETTEN, A., GRISONI, L., ANDRIOT, C., AND BARSKY, B. 2008. Geometrically exact dynamic splines. *Comput. Aided Des. 40*, 1 (Jan.), 35–48.

WARD, K., BERTAILS, F., KIM, T.-Y., MARSCHNER, S. R., CANI, M.-P., AND LIN, M. C. 2007. A survey on hair modeling: Styling, simulation, and rendering. *IEEE Transactions on Visualization and Computer Graphics 13*, 2 (Mar.), 213–234.

YIN, K., LOKEN, K., AND VAN DE PANNE, M. 2007. Simbicon: Simple biped locomotion control. In *ACM SIGGRAPH 2007 papers*, vol. 26 of *SIGGRAPH '07*.